

Design Optimization of 6-RUS Parallel Manipulator Using Hybrid Algorithm

Alaa Aldeen Joumah, Chadi Albitar

Higher Institute for Applied Sciences and Technology (HIAST), Damascus, Syria
E-mail: alaaaldeen.joumah@hiast.edu.sy, shadi.albitar@hiast.edu.sy

Received: 08 October 2017; Accepted: 17 November 2017; Published: 08 February 2018

Abstract—In this paper, we present a study on the design optimization of the 6-RUS Stewart platform using a hybrid algorithm. The geometric and kinematic models are calculated. The optimization problem is formulated after determining the design parameters and defining a set of cost functions related to the size of the workspace and to the indices of the kinematic and static performance, which are the global conditioning index (GCI) and the global stiffness index (GSI).

We started by studying the relation between the design parameters and the proposed cost functions, and then we invested the genetic algorithm to optimize each cost function separately. Moreover, we adopted a weighted cost function method to solve the Multi-Objective optimization problem.

The convergence performance of the genetic algorithm (GA) and the particle swarm optimization (PSO) were compared, where the PSO algorithm showed better performance. Based on this, a hybrid PSO–PS method was proposed and the results are highly competitive as we obtained better general convergence performance.

Index Terms—Stewart platform, RUS, optimization, performance indices, GA, PSO.

I. INTRODUCTION

Parallel manipulators are increasingly used in various industrial applications for their high capacity of load carrying, good dynamic properties, and high precision in positioning. These manipulators consist of several serial chains supporting a single mobile platform or an end-effector. Parallel manipulators can be classified based on degree of freedom, number of arms, order of joints in each chain and type of actuators. According to this, various 6-DOF parallel manipulators have been proposed. One of the most important 6-DOF parallel manipulators is 6-RUS manipulators that Revolute joint, Universal joint, and Spherical joint are used in each arm, respectively.

The most important advantage of 6-RUS manipulators is the low weight of movable parts because of installing the motors on the fixed platform. Therefore, bigger and cheaper electrical motors can be used. Moreover, these manipulators can be balanced statically [23].

Although this type of manipulators has some

disadvantages such as bending in connecting rods and complicated mechanical analysis. Also, because of the large number of chains and of the movement limitations of passive joints, such as spherical and universal joints, the workspace of the 6-RUS manipulators is restricted. In addition, the performance of parallel mechanisms is heavily dependent on their geometric parameters.

For all these reasons, the determination of the geometric parameters, i.e., the optimum design, of parallel mechanisms is relatively more important and difficult, and it has been attracting more and more researchers.

Optimizing the parallel manipulators has many difficulties. First, optimizing the parallel manipulators is a multi-objective and multi-criterion problem based on different performance specifications, and sometimes the considered specifications have diverse relationships. Another issue is that there are no direct relationships between the performance specifications and the structural parameters, and therefore, solving the optimization problems will have several responses.

The optimal design problem of parallel manipulators consists of determining a set of design parameters to guarantee an optimal criterion. Generally, some performance criterions, such as control accuracy (isotropy or dexterity), speed, payload capability, and stiffness may be involved in this process.

However, only limited studies have been proposed on the optimal design of the 6-RUS manipulators compared with the studies on the other parallel platforms with linear actuators (such as the 6-PUS platform). Stoughton and Arai presented a modification of the Stewart platform and optimized its design with respect to a weighted sum of dexterity and workspace volume [1]. Zhou proposed an optimal design method based on dynamic isotropy [2]. Su used the condition number of Jacobian matrix as an objective function to get accurate trajectory tracking [3]. Merlet proposed a general methodology for multi-criteria optimization of parallel manipulators based on interval analysis [4]. A. Taherifar proposed a solution for the kinematic optimization of Stewart platform for simulators using genetic algorithm which used a multi-objective cost function to maximize the workspace and to minimize the singularity condition within the workspace at the same time [5]. Gao *et al.* described the implementation of genetic algorithms and artificial neural networks as an

intelligent optimization tool for the dimensional synthesis of the spatial 6-DOF parallel manipulator. The multi-objective optimization (MOO) problem consisted of two functions: system stiffness and dexterity, which are derived according to kinematic analysis of the parallel mechanism [6]. C. Bonilla and A. Alexander proposed a solution for the optimization of the orientation workspace of the Stewart 6-RUS platform using the PSO algorithm [7]. Mirshekari *et al.* presented a structure comparison and optimal design of 6-RUS parallel manipulator based on kinematic and dynamic performances, where the 6-RUS manipulator structure is optimized using the modified multi objective Bees Algorithm [8].

Actually, the first step in the optimal design of the parallel platforms is determining the platform structure, i.e., its geometrical configuration. Three samples of the most well-known 6-RUS group manipulators are classified in [8], and these samples include Hunt type (Hunt, 1983), Hexa (Pierrot, 1990) and a manipulator named Zamanov type (Merlet, 2006) (Fig. 1).

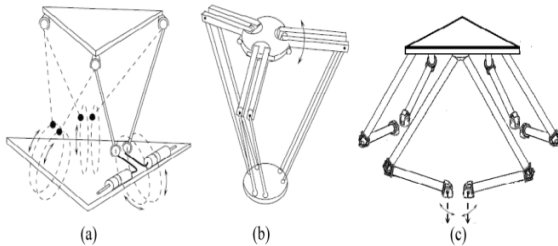


Fig.1. 6-RUS parallel manipulators (a): Hunt type (b): Hexa (c): Zamanov type.

In our research, we have adopted the study of the **Zamanov** type, for its relatively intermediate advantages, in order to improve its kinematic performance by achieving its optimal design.

The main objective of this study is to introduce a hybrid method that combines particle swarm optimization (PSO) and Pattern search method (PS) in the context of solving multi-objective optimal design problem under some equalities and inequalities constraints.

The rest of the paper is organized as follows. The proposed optimal design process is described in Section II. In Section III, our case study of 6-RUS manipulator is described and its geometric and kinematic models are recalled. Then, the corresponding multi-objective optimization problem is formalized in Sections (IV- V) by defining the set of cost-functions and determining the design parameters and the limitations of the platform. Section VI presents the results of the optimization problem; it discusses the results of GA, PSO and hybrid algorithm. Finally, Section VII summarizes our contributions as we propose some perspectives to this work.

II. OPTIMAL DESIGN PROCESS

The optimal design of the parallel platforms requires a set of mathematical models for the geometric and

dynamic characterization. Initially, we adopted a process that combines the modeling stage with the optimal design stage (Fig. 2).

The geometrical information resulting from the modeling stage forms an input to the design stage, while the information resulting from the design stage constitutes a feedback to the modeling stage. This loop is applied until the ideal design is achieved.

First, we have to define the problem to be solved. Depending on the objectives of the study, the applied steps may vary; it could be finding the geometric, kinetic, or dynamic models of the robot using different techniques. These models are important to apply high performance control algorithms, to improve stiffness, to increase payload or to improve force/torque capacity.. etc. The objective could be also finding the optimal design that aims at enhancing the performance indexes by adjusting the structural parameters, such as the dimensions of the platform. In the optimal design, several performance indices are involved such as stiffness, volume of workspace, transmission ratio and accuracy.

III. THEORETICAL STUDY

In this paragraph, we present the mathematical formula for the geometric and kinematic models of the studied platform.

A. Geometric modeling

The modeling stage starts by the geometric modeling which represents the relations between the location vector of the end-effector X and the joint coordinate vector q . Several methods and notations have been proposed to find the geometric model; the most widely used one is that of Denavit-Hartenberg [9]. However, this method is developed for simple serial-structured robot. Khalil and Klein have proposed a unified description of parallel and tree structured robots [10].

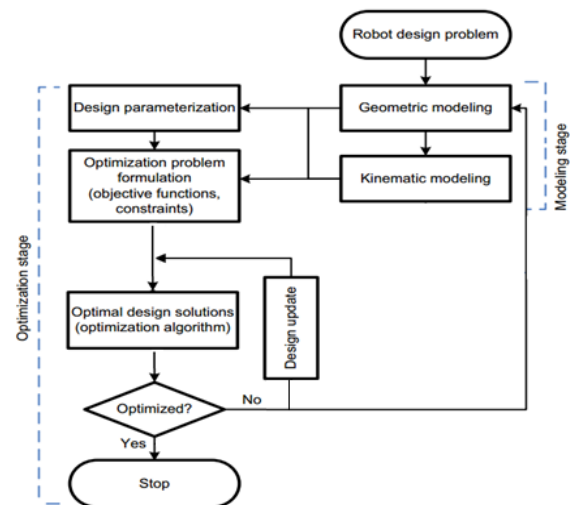


Fig.2. Work process (modeling stage and optimization design stage).

The direct geometric model (DGM) of the Stewart platform is defined as follows:

$$\begin{aligned} X &= f(q) \\ : X &= [p_x, p_y, p_z, \varphi, \theta, \psi] \\ q &= \theta_i = [\theta_1, \theta_2, \dots, \theta_6] \end{aligned} \quad (1)$$

The inverse geometric model (IGM) is also known as:

$$q = f^{-1}(X) \quad (2)$$

In general, finding the inverse geometric model in the case of parallel robots is easier than finding a direct geometric model. The IGM is based on relatively simple geometric relationships compared to the complex methods needed to find the DGM, which is generally based on numerical and iterative processes [11].

In our case, the inverse geometric model is determined in a geometric method according to the following relationships [12] (Fig. 3):

$$\begin{aligned} \theta_i &= a \tan 2(S_i, C_i) \\ S_i &\equiv \sin \theta_i = \frac{P_i Y_{C_i} + X_{C_i} \delta_i \sqrt{\Gamma_i}}{X_{C_i}^2 + Y_{C_i}^2} \\ C_i &\equiv \cos \theta_i = \frac{P_i X_{C_i} - Y_{C_i} \delta_i \sqrt{\Gamma_i}}{X_{C_i}^2 + Y_{C_i}^2} \\ &: \left(\begin{array}{l} \Gamma_i = X_{C_i}^2 + Y_{C_i}^2 - P_i \\ P_i = \frac{L_i^2 + \ell_1^2 - \ell_2^2}{2\ell_1}, L_i = \|A_i C_i\| \end{array} \right) \end{aligned} \quad (3)$$

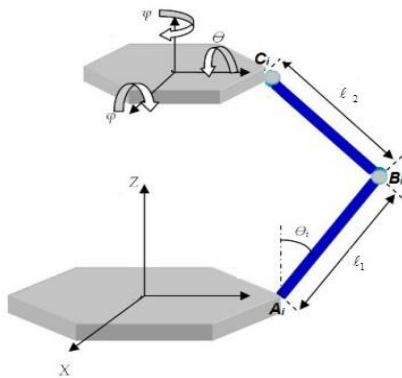


Fig.3. Modeling the 6-RUS mechanism.

B. Kinematic modeling

Kinematic model aims to find the relation between the end-effector velocity and the joint velocities. Kinematic model could be written using the Jacobin matrix \mathbf{J} which appears in calculating the derivation of the geometric model. It gives the differential variations of the operational coordinates \dot{X} in terms of the differential variations of the joint coordinates \dot{q} . For parallel manipulator, the key concept is to “break” the parallel manipulator into “simple” serial chains taking into consideration the mechanical constraints.

$$\dot{X} = J(q)\dot{q} \quad (4)$$

The Jacobian matrix has multiple applications in robotics as it facilitates the calculation of singularities and of the dimension of accessible operational space of the robot [13]. In static force model, we use the Jacobian matrix in order to calculate the forces and torques of the actuators in terms of the forces and moments exerted on the end-effector. The static model is essential in structural analysis as well as in formulating the optimal design problem.

For the 6-RUS platform, the kinematic model is derived from the derivation of the inverse geometric model [12] according to the following relationships:

$$\begin{aligned} J_q \dot{q} &= J_x \dot{X} \\ \Leftrightarrow \dot{q} &= J \dot{X} : J = (J_q)^{-1} J_x \end{aligned} \quad (5)$$

$$\text{Where: } \dot{X} = \begin{bmatrix} {}^B \dot{x}_p \\ \omega_p \end{bmatrix}, \quad \dot{q} = \dot{\theta} = [\dot{\theta}_1, \dots, \dot{\theta}_6]$$

$$\begin{aligned} J_x &= \begin{bmatrix} \lambda_1^T & ({}^B c_1 \times \lambda_1)^T \\ \lambda_2^T & ({}^B c_2 \times \lambda_2)^T \\ \vdots & \vdots \\ \lambda_6^T & ({}^B c_6 \times \lambda_6)^T \end{bmatrix} \\ J_q &= \text{diag} [\lambda_{q1}, \lambda_{q2}, \dots, \lambda_{q6}] \end{aligned} \quad (6)$$

$$\text{Where: } \begin{cases} \lambda_i = ({}^B C_i - {}^B B_i) \\ \lambda_{qi} = \lambda_i^T {}^B R_{(i)} \begin{bmatrix} -\ell_1 \sin(\theta_i) \\ \ell_1 \cos(\theta_i) \\ 0 \end{bmatrix} \end{cases}$$

IV. FORMULATION OF THE OPTIMAL DESIGN ISSUE

The optimal design aims to reach the optimal geometric configuration according to objective functions and geometric constraints. In general, optimal design problem is described as follows:

What are the best values of the design parameters which ensure that the set of performance criteria is met under the set of imposed geometric constraints?

Mathematically, this issue is formulated as in the following relationships:

$$\begin{aligned} \text{Find a Vector } P^* &= [p_1^*, p_2^*, \dots, p_n^*] \\ \text{that Minimize: } F(P) &= [f_1(p), f_2(p), \dots, f_k(p)]^T \\ \text{with } g_m(P) &\leq 0 \quad (m \text{ inequality constraints}) \\ \text{and } h_\ell(P) &= 0 \quad (\ell \text{ equality constraints}) \end{aligned} \quad (7)$$

In this formulation, p is the vector of the design parameters and F is the vector of the considered performance criteria within the optimal design issue, while g and h are vectorial functions involved respectively in the inequality and equality constraints,

which are imposed on the parallel platform.

Problems without any constraint g and h are called unconstrained while the others are constrained.

The following sections describe the design parameters of the Stewart platform 6-RUS and the constraints of the optimization problem.

A. Design parameters:

To formulate an optimization problem that can optimize the kinematic performance indices, appropriate design parameters must be selected. The design parameters must influence the objective function directly or indirectly. Design parameters can be regarded as free variables, as they can be assigned any value within a given range or subset by the optimization algorithm. This range or subset is mathematically formulated as constraints. The design parameters should be independent of each other as far as possible. In general, the number of design parameters determines the degree of freedom for the optimization problem. Optimization complexity is increased according to the degree of freedom of the problem. Thus, the number of design parameters should be kept as low as possible.

The Stewart-Gough platform geometry was defined with two coplanar semi regular hexagons; the first corresponds to the base hexagon and the second to the movable platform (Fig. 4).

In our case we need at least six basic geometric parameters, which include the lengths of the arms of the platform and the dimensions of its bases.

Therefore, the vector of the design parameters can be defined as follows:

$$P = [\ell_1, \ell_2, d_{b1}, d_{b2}, d_{p1}, d_{p2}] \quad (8)$$

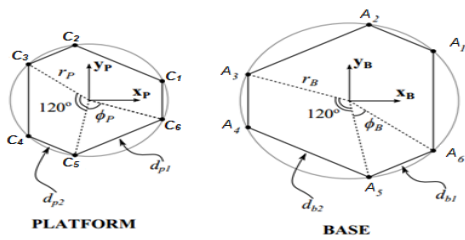


Fig.4. Dimensions of the lower and upper bases.

B. The constraints:

While designing a practical manipulator, the physical constraints of the kinematic chains, such as limits of the rotary actuators, arm interferences and limitations on the passive joints should be considered.

As for the constraints on the rotary actuators in the platform, each of them has a specific range of work. In our case, we considered a range of work as follows:

$$(\theta_{i\min} = -40^\circ) \leq \theta_i \leq (\theta_{i\max} = 100^\circ) \quad (9)$$

For the constraints of spherical and universal joints, the limits on the cone angle of a spherical joint should be considered because of the difficulties in manufacturing of

spherical joints. In our case, each joint within the platform has a specific range of work, as follows:

$$\begin{aligned} |\max(\alpha_{i1})| \leq 45^\circ & : \alpha_{i1} = \cos^{-1}(n_{\ell_{2i}} \cdot n_p) \\ |\max(\alpha_{i2})| \leq 30^\circ & : \alpha_{i2} = \cos^{-1}(n_{\ell_{1i}} \cdot n_{\ell_{2i}}) \end{aligned} \quad (10)$$

Where α_{i1} expresses the angle of the universal joint that is formed between the vertical vector on the upper base n_p and the unit vector according to the ineffective arm $n_{\ell_{2i}}$, while α_{i2} denotes the angle of the spherical joint.

The arm interference is also considered in the evaluation of the minimum distance between two arms in order to avoid collision.

In addition, the dimensionality of the design parameters is taken also into consideration. Each parameter of the design parameter P takes its value within a specific range of work. Table 1 shows the limitations that we considered in our case.

Table 1. Upper and lower limits of the design parameters.

d_{b1}	d_{b2}	d_{p1}	d_{p2}	ℓ_1	ℓ_2	Parameter
20	20	10	5	5	10	Lower limits (cm)
35	30	30	15	20	35	Upper limits (cm)

V. COST FUNCTIONS

In this paragraph, we present the mathematical formula for the set of famous cost functions, special for parallel platforms.

A. Workspace

The workspace is defined as the set of space operational points that the terminal body of robot can access. The workspace of a parallel manipulator is one of the most important criterion which reflects its capacity. So it is necessary to analyze the shape and the size of the workspace for enhanced applications of parallel manipulators.

In general, the workspace is defined for the parallel platforms using the geometric methods or the discretization methods.

Within the recursive optimization algorithm, the workspace computation process must be run numerous times. Hence, importance of the efficiency of the algorithm and the reduction of total execution time of the workspace computation algorithm are highlighted.

We chose to rely on the numerical search method due to its efficiency and speed of implementation. This method is based on Cartesian coordinates to find the boundaries of the workspace and its size ($V_{workspace}$). Fig. 5 presents the result of implementing of the discretization algorithm in MatLab environment in order to determine the translation workspace of the studied platform with Constant Orientation [$\phi = 0^\circ, \theta = 0^\circ, \psi = 0^\circ$].

This study was carried out for a set of the geometric parameters shown in table 2 which correspond to the

values of a real experimental platform.

Table 2. Values of the design parameters adopted in the tests.

d_{b1}	d_{b2}	d_{p1}	d_{p2}	ℓ_1	ℓ_2	parameter
26	20	30	6	11	25.5	Value (cm)

B. The Kinetic-Static performance indices:

In general, the Kinetic-Static performance indices are considered as the most important criteria during the optimal design process of robotic platforms, and in particular, for the parallel platform [14].

These indices include the Kinematic and Static Dexterity. The Kinematic Dexterity is defined as the ability of the terminal body of the platform to move accurately and with high repetition within the workspace, whereas the Static Dexterity is defined as the terminal body's ability to apply force and power in all directions within the platform's workspace. Dexterity is an important issue for design, trajectory planning, and control of manipulators. These coefficients are based mainly on the Jacobian matrix J , which is used in the linear conversion of both velocity and force within the mechanism. Thus, this matrix contains useful information regarding the transformation between the associative and operational spaces.

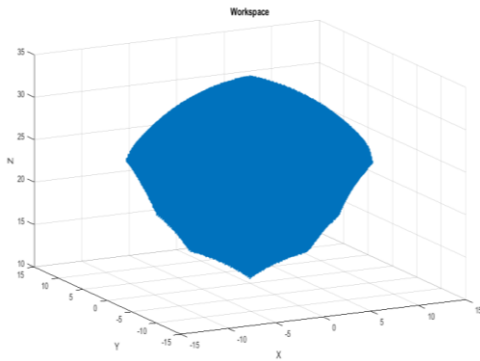


Fig.5. Translation workspace with Constant Orientation.

The following section describes the most important kinematic performance indices of the Stewart platform 6-RUS which are considered in this research. The two Global performance indices GCI, GSI are widely adopted for the evaluation of global behavior of manipulators.

a. Global Conditioning Index (GCI):

This parameter is very important during the process of the optimal design of parallel platforms, where the value of Kinematic Dexterity is measured through it. The definition of this parameter depends on the conditional number of the jacobian matrix, which is calculated by taking the ratio between the maximum and minimum value of the singular values of the matrix J :

$$\kappa(J) = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{11}$$

Kinematic dexterity is determined at a particular pose of the end-effector within the workspace, and calculated by taking the inverted value of the conditional number in that position [15]. Thus, the value of local kinematic dexterity varies from zero (Which corresponds to a singular position) to one (Which corresponds to an ideal position).

Fig. 6 shows changes in the value of the local kinematic dexterity (LCI) of the studied platform (6-RUS) within the XY plane (where $Z=24.5$ cm), using the values of the design parameters shown in table 2.

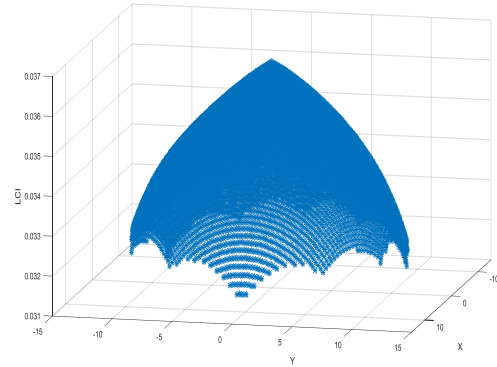


Fig.6. Local kinematic dexterity changes within XY plane.

We notice that this parameter always reaches its maximum value in the center of XY plane, where this value starts decreasing as we move away from this center in all directions.

The Kinematic dexterity is determined over the whole workspace of the parallel platform by GCI coefficient, which is defined according to the following relationship [16]:

$$\eta_{GCI} = \frac{\int_W 1/\kappa(J) dw}{\int_W dw} \tag{12}$$

Where W denotes the workspace of the platform.

The GCI coefficient may be defined in the discrete description as follows:

$$\eta_{GCI} = \frac{1}{n} \sum_{i=1}^n \frac{1}{\kappa_i(J)} \tag{13}$$

Where n denotes the number of the active points forming the workspace of the platform.

During the optimal design process of the platform, we aim to reach the highest possible value of the GCI coefficient to obtain an ideal isotropic mechanism on its whole workspace.

b. Global stiffness index (GSI):

The manipulator stiffness affects the dynamics and the position accuracy, for which stiffness is considered also as an important criterion during the optimal design process of the parallel platforms. Practically, the terminal

body of the parallel platform is subjected to distortions due to the influence of external forces. These distortions depend on the stiffness of the mechanism and on the value of external forces. Theoretically, in spatial coordinate system, the stiffness matrix K relates the external forces vector τ at the moving platform to the output displacement vector D according to the following platform stiffness model:

$$\tau = K.D \quad (14)$$

Considering a unit vector of applied external forces, the maximum and minimum values of the terminal body deformities vector are determined using the following relationships:

$$\|D_{\max}\| = \sqrt{\max(|\lambda_{Di}|)} \quad \& \quad \|D_{\min}\| = \sqrt{\min(|\lambda_{Di}|)} \quad (15)$$

Where λ_{Di} denote the Eigen values of the matrix $(K^{-1})^T K^{-1}$.

The maximum and minimum values of deformities constitute the Deformation ellipsoid, whose main axis is based on the directions of the Eigen vectors of the matrix $(K^{-1})^T K^{-1}$. Depending on these maximum and minimum values of deformities, GSI parameter can be defined on the whole workspace of the mechanism using the following relation:

$$\eta_{GSI} = \frac{\int_W \left(\frac{\|D_{\min}\|}{\|D_{\max}\|} \right) dw}{\int_W dw} \quad (16)$$

The aim is to minimize the values of deformities of the platform by increasing the value of GSI coefficient [17].

VI. PRACTICAL RESULTS

After formulating the optimization problem, the next step is to find an optimal design solution using an appropriate algorithm.

In this paragraph, we present at first the results of the study on the effect of the geometric design parameters on the considered cost functions, then we present here the practical results of the single-objective and the multi-objective optimization problem.

A. The Effect of design parameters on Cost functions:

We present here the results of the study on the effect of the geometric design parameters on two of the considered cost functions, which are the size of the workspace and GCI coefficient. We study the effect of the lengths of the arms (ℓ_1, ℓ_2) , and the effect of the difference between the dimensions of the platform bases. We chose to fix the dimensions of the lower base and adjust the dimensions of the upper base (d_{p1}, d_{p2}) . Fig. 7 shows the effect of

previous parameters on the volume of the translation workspace of the Stewart platform 6-RUS. We notice the direct effect of the arms (ℓ_1, ℓ_2) on the size of the workspace as the size increases by increasing their value, with a greater influence of the parameter ℓ_1 (fig.7-(b)). The effect of the dimensions of the upper base is different, the size of the workspace increases as the value of the parameter d_{p2} increases. However, it reaches its maximum near the central value of d_{p1} (near 20cm) (fig.7-(a)).

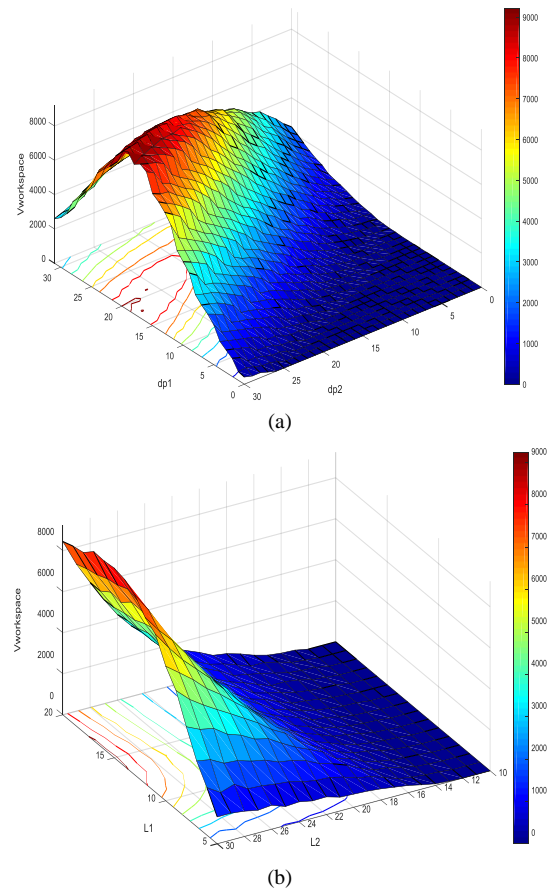


Fig. 7. The effect of design parameters on the volume of the workspace.

We also studied the effect of the design parameters on the value of GCI (Fig. 8). The inverse effect of the value of the parameter d_{p2} on the value of the GCI is obvious.

In contrast, the parameter d_{p1} has a significant effect within a specific range [13-16 cm], where we obtain the maximum value of the studied coefficient (GCI) (fig.8-(a)). For the effect of the arms of the platform, we notice the lack of significant impact of the parameter ℓ_1 on GCI coefficient, while the value of the parameter ℓ_1 as a great effect within the range [18-23 cm] as the value of GCI reaches its maximums and begins to decrease as it moves away from this area (fig.8-(b)).

B. Single-objective optimization

As a result, the optimal design problem of the Stewart

platform 6-RUS is described as follows:

$$\begin{aligned}
 & \text{Maximize: } V_{\text{workspace}} \rightarrow f_1(p) = -V_{\text{workspace}} \\
 & \text{Maximize: } GCI \rightarrow f_2(p) = -\eta_{GCI} \\
 & \text{Maximize: } GSI \rightarrow f_3(p) = -\eta_{GSI} \\
 \\
 & \text{Over: } P = [\ell_1, \ell_2, d_{b1}, d_{b2}, d_{p1}, d_{p2}] \\
 & \text{Subject to:} \\
 & g_1: \quad -40^\circ \leq \theta_i \leq 100^\circ \\
 & g_2: \quad |\alpha_{i1}| \leq 45^\circ \quad i = 1, \dots, 6 \\
 & g_3: \quad |\alpha_{i2}| \leq 30^\circ \quad i = 1, \dots, 6 \\
 & g_4: \quad (5 \text{ cm}) \leq \ell_1 \leq (20 \text{ cm}) \\
 & g_5: \quad (10 \text{ cm}) \leq \ell_2 \leq (35 \text{ cm}) \\
 & g_6: \quad (20 \text{ cm}) \leq d_{b1} \leq (35 \text{ cm}) \\
 & g_7: \quad (20 \text{ cm}) \leq d_{b2} \leq (30 \text{ cm}) \\
 & g_8: \quad (10 \text{ cm}) \leq d_{p1} \leq (30 \text{ cm}) \\
 & g_9: \quad (5 \text{ cm}) \leq d_{p2} \leq (15 \text{ cm})
 \end{aligned} \tag{17}$$

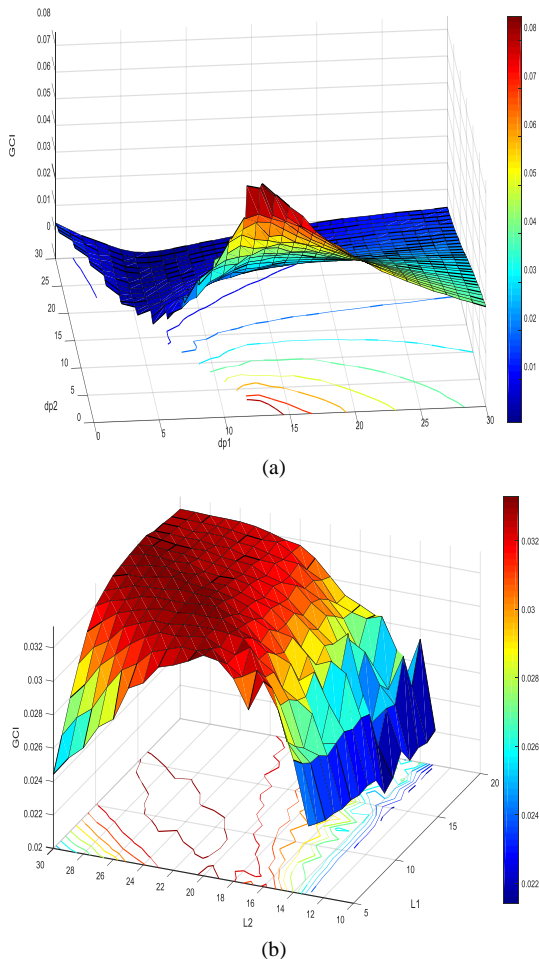


Fig.8. The effect of design parameters on the GCI coefficient.

An optimization method should be selected to obtain the optimum values of the design parameters. However, it is hard to solve this problem analytically. Furthermore, derivative-based methods are not appropriate because of the non-smooth behavior of these objective functions. In

addition, they are unable to escape local minima. Therefore, intelligent and direct search methods are suitable such as Genetic Algorithm (GA), Particle Swarm optimization (PSO) or Pattern Search (PS) methods or their combination.

In this work, we invested the most famous algorithm (the GA) to optimize each cost function separately, and we aim at maximizing the three proposed objective functions in the optimization process. GAs are heuristic search algorithms based on the mechanism of natural selection and natural genetics initially proposed by Holland [18]. GAs are high performance and robust optimization methods used to solve engineering problems.

The problem was modeled in MatLab environment, with the formulation of a set of cost functions. For each cost function separately, the genetic algorithm was applied according to the parameters described in table 3 and table 4 presents the obtained results.

Table 3. Genetic algorithm parameters.

Setting	Parameter
200	Population size
Real	Encoding type
Stochastic	Selection strategy
Scattered	Crossover type
Adaptive	Mutation type
0.9	crossing probability
0.1	probability of mutation

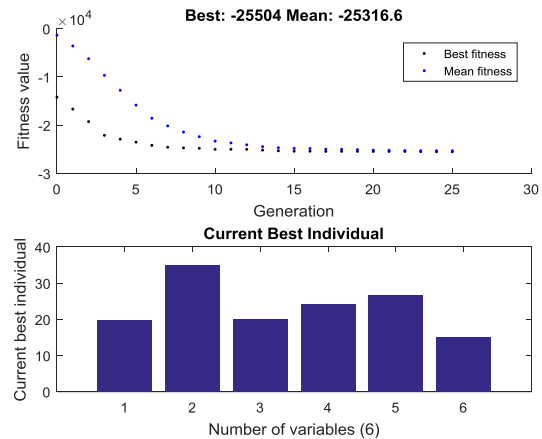


Fig.9. Results of convergence of the genetic algorithm for the size of workspace.

The $V_{\text{workspace}}$ is maximized in the first cost function. The evolution of workspace size as a function of generations is shown in Fig. 9. The maximum workspace volume is 25504 cm³, where the genetic algorithm converged to this optimal solution within 25 generations, and the corresponding design vector are shown in table 4. The obtained values of the design parameters meet the results of our study on the effect of design parameters on cost functions (in previous paragraph). As for the GCI coefficient, the obtained maximum value was (GCI=0.0772), and the algorithm converged to the optimal solution in 30 generations. The evolution of GSI

coefficient as a function of generations is shown in Fig. 10. The best value was (GSI=0.008) reached in 27 generations and the corresponding design vector are shown in table 4.

We notice in table 4 the differences between the resulting optimal solutions for each cost function. The optimal solution that achieves a greater size of workspace decreases the GCI to 31.8% of its maximum value, as well as the GSI to 8.1% of its maximum value. Therefore, there is a conflict in the resulting solutions which means that there is no optimal common solution for all these cost functions.

C. Multi-objective optimization

The principle of multi-objective optimization is different from single-objective optimization. In the single-objective optimization case, the solution of the optimization aims to obtain the best solution over all other alternatives. In the case of multi-objective optimization, there is not necessary a best solution due to conflicts among all objective functions. Therefore, the

result of the multi-objective optimization is a set of solutions which are called non-determinant or Pareto-optimal solutions when an improvement in one objective requires a degradation of another.

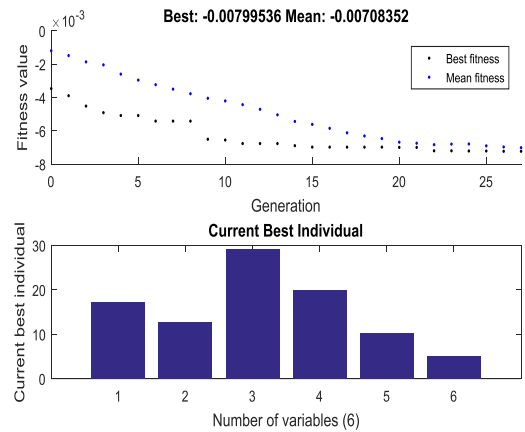


Fig.10. Results of convergence of the genetic algorithm for GSI coefficient.

Table 4. Results of single-objective optimization.

ℓ_1 (cm)	ℓ_2 (cm)	d_{b1} (cm)	d_{b2} (cm)	d_{p1} (cm)	d_{p2} (cm)	Single-Objective
19.79	34.99	20.09	24.05	26.77	14.97	Vworkspace
14.99	25.63	31.79	20.74	12.65	5.02	GCI
17.21	12.76	29.16	20.00	10.31	5.10	GSI

In order to realize a multi-objective optimization, we adopted the weighted sum method which, according to Arora [19], is the most commonly used method. Mathematically the multi-objective cost function is formulated using the weighted sum method by:

$$C = \sum_{j=1}^m \omega_j \cdot I_j \quad : \left(\sum \omega_j = 1 \right) \quad (18)$$

The weights W_j can be changeable according to the user demands.

The optimal design parameters are determined by finding the global minimum of the weighted sum function C, using numerical algorithms. Optimization with regard to multiple objectives often requires normalization of the individual objective functions to obtain similar order of magnitudes.

For the problem discussed in this paper, we define the following weighted sum cost function:

$$C = \omega_1 \left(\frac{V_{workspace}}{V_{workspace,max}} \right) + \omega_2 \left(\frac{GCI}{GCI_{max}} \right) + \omega_3 \left(\frac{GSI}{GSI_{max}} \right) \quad (19)$$

Where $\left(\omega_1, \omega_2, \omega_3 \in [0,1], \sum_{j=1}^3 \omega_j = 1 \right)$

The variation ranges of the proposed cost functions are

quite different. So we used its maximum values to normalize the functions between zero and one. These three maximum values ($V_{workspace,max}$, GCI_{max} , GSI_{max}) can be calculated by performing single-objective optimization for each function as we did in the previous section. Choosing the values of the weights W_j , in order to characterize a particular performance of the platform, is quiet tricky as the final result depends strongly on that choice.

Optimization of manipulator design is a nonlinear problem with nonlinear constraints, and often with one or more integer variables. The optimization can be achieved in various ways, from basic grid search, where the objective function is simply calculated for a set of variable values, to gradient based search, where a gradient is found for each step, providing a search direction for the next step, and to more or less advanced stochastic methods such as Monte Carlo simulation, GA and PSO.

In this work, we adopted two algorithms: the genetic algorithm GA and the PSO algorithm, and the Multi-objective cost function problem was solved for several values of weights W_j , using both algorithms in order to compare between their performances.

PSO has become increasingly popular in the field of robotic manipulators design as it emulates the behavior of a flock of birds or a swarm of bees [20] where each individual in the population is searching through the space for the best possible solution.

In MatLab (R2016a) environment, using a computer

with Intel (R) Core (TM) i5-3450 CPU @ 3.10GHz (8GB), the problem is practically formulated using the weighted sum cost function C , for 4 different cases of weights W_j as follows:

1. $\omega_1 = \omega_2 = \omega_3 = 1/3$
2. $\omega_1 = 1/2, \omega_2 = \omega_3 = 1/4$
3. $\omega_2 = 1/2, \omega_1 = \omega_3 = 1/4$
4. $\omega_3 = 1/2, \omega_1 = \omega_2 = 1/4$

A comparison of the two algorithms (GA, PSO) requires the selection of identical parameters. Table 5 shows the used PSO algorithm parameters, which are similar to the genetic algorithm parameters shown in table 3 .

In (table 6 & table 7), we present the optimal values of the design parameters P , which were reached for each of the four studied cases of the weights W_j . In addition, the minimum value of the weighted cost function was

calculated too. We also show the number of iterations necessary to converge towards the optimal solution as well as the time needed to implement them.

We note that the results have been reached using Parallel Computing in MatLab, in order to exploit the full capabilities of the computer in the practical implementation of the algorithm.

Table 5. PSO algorithm parameters.

Setting	Parameter
200	Swarm Size
Real	Encoding type
Stochastic	Selection strategy
Scattered	Crossover type
0.25	Min Neighbours Fraction
1.49	Social Adjustment Weight(C1,C2)
1.1	Inertia (W)

Table 6. Results of multi-objective problem using the GA.

ℓ_1 (cm)	ℓ_2 (cm)	d_{b1} (cm)	d_{b2} (cm)	d_{p1} (cm)	d_{p2} (cm)	Cost Function	Gen	Time (s)	Multi-Objective
19.99	24.01	30.08	20.03	12.43	5.03	-0.6067	43	1:49:40	First case
19.71	34.93	20.04	25.61	28.19	14.90	-0.5932	26	0:80:35	Second case
16.29	27.13	29.18	20.11	10.21	5.01	-0.7216	40	1:47:52	Third case
16.74	30.90	32.57	20.28	10	5	-0.7249	34	1:34:48	Fourth case

Table 7. Results of multi-objective problem using the PSO.

ℓ_1 (cm)	ℓ_2 (cm)	d_{b1} (cm)	d_{b2} (cm)	d_{p1} (cm)	d_{p2} (cm)	Cost Function	Gen	Time (s)	Multi-Objective
20	30.94	35	20	10.05	5	-0.6346	43	1:43:32	First case
20	35	20	20	22.34	15	-0.6087	26	0:76:23	Second case
18.58	30.58	35	20.01	10	5	-0.7479	40	1:42:36	Third case
18.57	30.59	35	20	10	5	-0.7380	34	1:27:04	Fourth case

Fig. 11 to Fig. 14 show a comparison between the performances of the two algorithms (GA & PSO) in their convergence towards the optimal solution. The evolution of the weighted cost function C (for the first studied case of weights W_j) as a function of generations is shown in figure 11.

Through the set of previous results and the graphs of both algorithms (for the 4 studied cases) we can state the following:

- For the same number of generations for both algorithms, the PSO algorithm always needs less execution time than the GA algorithm.
- The PSO algorithm has better performance as it has a faster convergence to optimal solution than genetic algorithm.

- The PSO algorithm reaches a better solution (a more optimal) than the one reached by the genetic algorithm.

Thus, the performance of the PSO algorithm outperforms the performance of the GA algorithm in solving the multi-objective optimization problem within this research, with regards to time consumption and accuracy. In some respects, PSO is similar to continuous GA but PSO has some advantages over GA [21]. PSO does not require extra operations such as crossover, and it has fewer parameters to adjust. In addition, convergence of PSO to optimum region for cost functions, with many variables, is better and faster than GA and constraining the variables in PSO is easier. For these reasons, we chose PSO algorithm to solve our optimization problem.

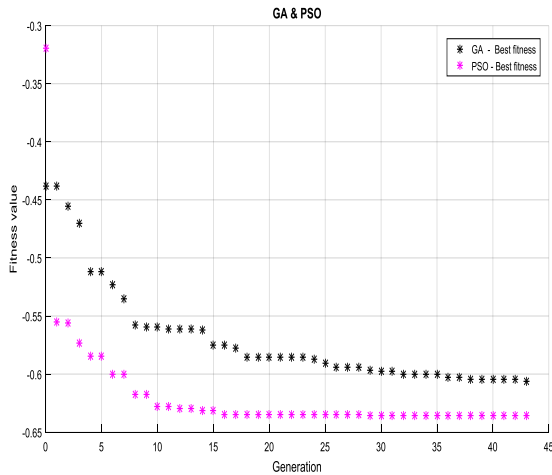


Fig.11. Comparison of the (GA & PSO) in the first case.

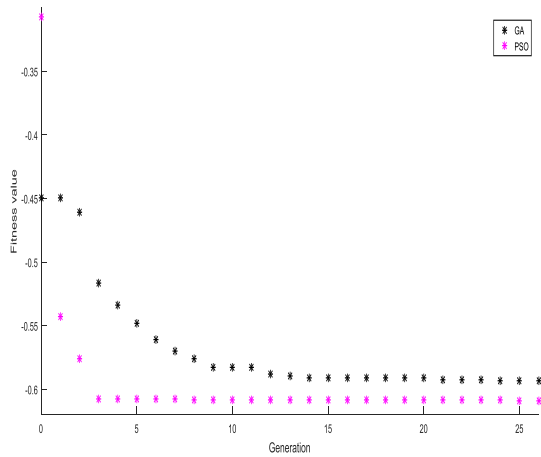


Fig.12. Comparison of the (GA & PSO) in the second case.

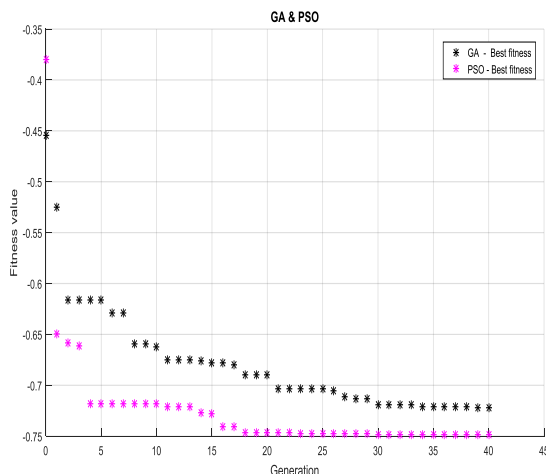


Fig.13. Comparison of the (GA & PSO) in the third case.

A common problem to all optimization techniques, including stochastic optimization algorithms, is that they do not generally guarantee convergence to the global optimum[22], and the random behavior of this algorithms leads to different solutions for various runs. So, the problem now is to make sure that the PSO algorithm

solution is the best global minimum solution. This problem can be solved by using the hybrid optimization algorithm.

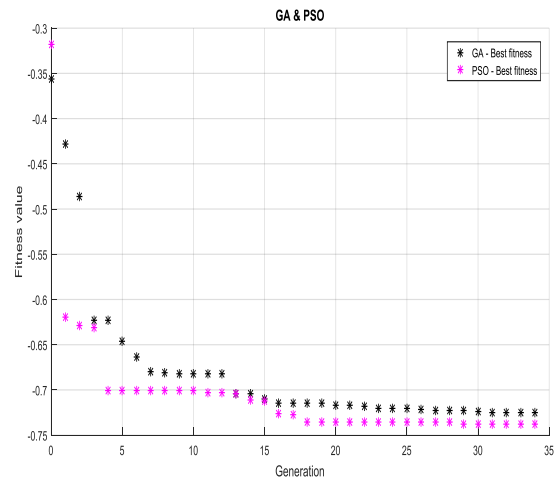


Fig.14. Comparison of the (GA & PSO) in the fourth case.

Thus, the performance of the PSO algorithm outperforms the performance of the GA algorithm in solving the multi-objective optimization problem within this research, with regards to time consumption and accuracy. In some respects, PSO is similar to continuous GA but PSO has some advantages over GA [21]. PSO does not require extra operations such as crossover, and it has fewer parameters to adjust. In addition, convergence of PSO to optimum region for cost functions, with many variables, is better and faster than GA and constraining the variables in PSO is easier. For these reasons, we chose PSO algorithm to solve our optimization problem.

A common problem to all optimization techniques, including stochastic optimization algorithms, is that they do not generally guarantee convergence to the global optimum[22], and the random behavior of this algorithms leads to different solutions for various runs. So, the problem now is to make sure that the PSO algorithm solution is the best global minimum solution. This problem can be solved by using the hybrid optimization algorithm.

As a result of the large complexity of the proposed cost functions, and due to the possibility of local minimum solutions, it is necessary to propose reliable strategies to ensure reaching the global optimal solution, taking into account the efficiency of fast convergence towards the desired solution.

To this end, we searched a solution based on the coupling between two or more different optimization algorithms which have complementary characteristics. This is what is so-called hybrid algorithms. It is common to find hybrid algorithms involving an algorithm of stochastic type used to cover the entire search space to identify the region where the global minimum may be found, and a deterministic algorithm with mathematical reasoning able to quickly reach the minimum, since the region has been identified. This type of strategy improves reliability compared to methods of non-linear programming as it is more likely able to find the global

minimum, and besides, it increases efficiency compared with pure stochastic algorithms. However, combining stochastic and deterministic methods into a hybrid solution can prevent both inability to converge, and premature convergence tendencies.

In this work, we proposed the combination between PSO algorithm and the generalized pattern search PS algorithm to achieve the desired optimal global solution. Hence, a pattern search function in our hybrid method will start from the final solution obtained by PSO, so their combination leads to a unique solution.

PS techniques are a class of direct search methods for optimization. Like genetic algorithm, PS is often useful for global optimization. It is suitable for non-smooth or discontinuous objective functions. PS algorithm searches a set of points, called a mesh, around the current point. The mesh is created by adding the current point to a scalar multiple of a set of vectors called a pattern. If the algorithm finds a point within the mesh that improves the objective function, the new point becomes the current point at the next iteration. The search terminates after a minimum mesh size is reached.

Based on the optimal results obtained by the PSO algorithm, we implemented a PS algorithm, whose parameters are shown in table 8. The problem of multi-objective optimization is formulated using the weighted sum cost function C , for the 4 different cases of weights W_j .

Table 9 shows the results obtained by the hybrid algorithm (PSO - PS).

The following figures (Fig. 15 to Fig. 18) show the results of the pattern search algorithm PS, for the 4 studied cases of the weighted cost function. In each of these figures, we present first a graph of the algorithm convergence towards the optimal solution as a function of iterations, as well as the evolution of the mesh size in the second graph, and finally we present the obtained optimal values of the design parameters.

Table 8. PS algorithm parameters.

Setting	Parameter
GPS Positive basis 2N	Poll Method
on	Complete Poll
MADS Positive basis 2N	Search Method
on	Complete Search
1	Initial mesh size
2	Expansion Factor
0.5	Contraction Factor
200	Max Iteration
10 ⁻⁶	Mesh Size to stop

Table 9. Results of multi-objective Optimal Design problem using the hybrid algorithm (PSO+PS).

ℓ_1 (cm)	ℓ_2 (cm)	d_{b1} (cm)	d_{b2} (cm)	d_{p1} (cm)	d_{p2} (cm)	Cost Function	Gen	Time (s)	Multi-Objective
20	30.94	34.97	20	10.06	5	-0.6350	31	0:14:27	First case
20	35	20	20	22.34	14.99	-0.6091	62	0:25:11	Second case
18.58	30.58	34.98	20	10.03	5	-0.7493	57	0:27:25	Third case
18.57	30.59	34.99	20	10	5	-0.7381	26	0:15:18	Fourth case

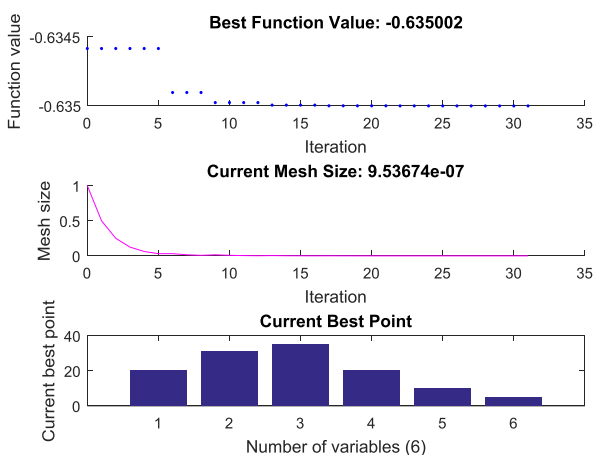


Fig.15. Results of PS algorithm for the first case.

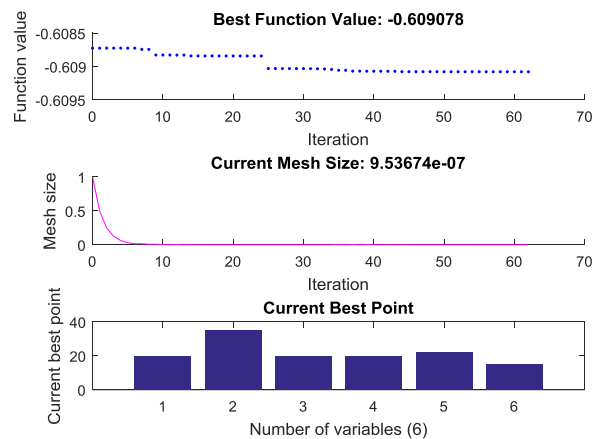


Fig.16. Results of PS algorithm for the second case.

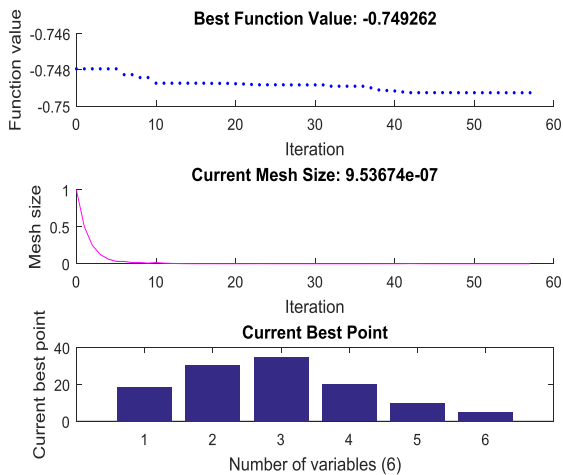


Fig.17. Results of PS algorithm for the third case.

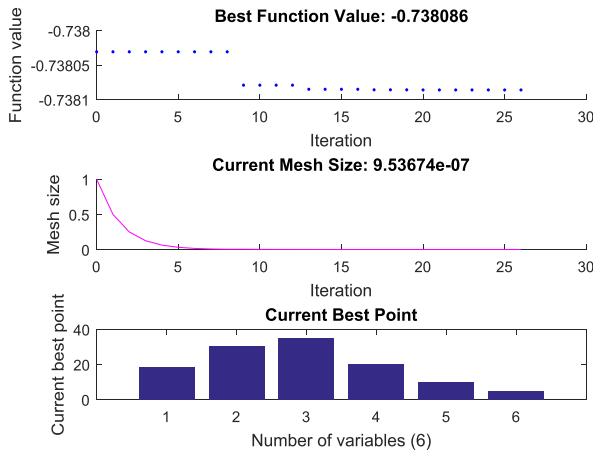


Fig.18. Results of PS algorithm for the fourth case.

Through the previous set of results for the hybrid algorithm, and compared with the results of the PSO algorithm alone, we note the following:

- The value of the global minima solution of weighted cost function, within the results of the hybrid algorithm is better than for the result of the PSO algorithm; this proves the importance of the hybrid algorithm by ensuring that the global optimal solution is reached.
- The values of the optimal design parameters resulting from the both algorithms (hybrid, PSO) are very close, which confirms that the PSO algorithm is also very close to the global solution while the hybrid algorithm reached this global solution accurately.
- The convergence of the PS algorithm to the global solution is faster, and this is because it starts from the final solution obtained by PSO, which is very close to the optimal solution.
- The proposed hybrid method has led to the reduction of total execution time of the algorithm compared to other methods.
- The proposed (PSO-PS) technique has overcome an important drawback of the PS methods which is

the need to provide a suitable starting point. This shortcoming of the PS methods was highlighted in the previous work of the authors as it makes any optimization method relying on a good choice of the initial point possibly more susceptible to getting trapped in local minima.

Thus, the efficiency of the proposed hybrid algorithm (PSO-PS) to solve the multi-objective optimization problem in this research is demonstrated by ensuring that the only optimal global solution of the studied problem is achieved.

In this paper, the hybridization of PS method and PSO are incorporated in the optimization process in order to look for the global optimal solution in solving the multi-objective cost function and determining the design variables with minimum computational CPU time.

VII. CONCLUSION

This work tackles the optimal design of a 6-DOF Stewart platform (6-RUS Zamanov Type) with respect to multi-objectives based upon the hybrid (PSO-PS) method. We have modeled the studied platform geometrically and cinematically, and the reachable workspace of the platform is generated using numerical search method. The Optimal design problem was formulated by determining the set of design parameters (the dimensions of the platform bases and the arms $P = [\ell_1, \ell_2, d_{b1}, d_{b2}, d_{p1}, d_{p2}]$). Moreover, we defined a set of required cost functions related to the size of the workspace, and to the two Global Kinetic-Static performance indices (the GCI and GSI coefficients), where we have studied the effect of the design parameters on this studied cost functions.

The single-objective optimization design problem was solved through the GA algorithm for each cost function separately, and there was a conflict in the obtained optimal solutions. The Multi-objective optimization problem was solved through the weighted sum method, where the problem was solved for several values of weights W_j (in the weighted cost function) using both algorithms: the genetic algorithm GA and the PSO algorithm. As the PSO algorithm showed better performance, a hybrid algorithm that combines (PSO-PS) was proposed in order to achieve the unique global optimal solution and to obtain more accurate results. The results are valuable in designing these kinds of manipulators under different conditions, where the proposed hybrid algorithm showed the best performance with regards to time consumption and accuracy.

The proposed algorithm and the described approach are absolutely generic and can be used with different objective functions and constraints. Moreover, this approach can be extended to other types of parallel manipulators.

REFERENCES

- [1] R. Stoughton and T. Arai, "A modified Stewart platform

- manipulator with improved dexterity," *Robotics and Automation, IEEE Transactions on*, vol. 9, pp. 166-173, 2002.
- [2] H. Z. Jiang, J. F. He, and Z. Z. Tong, "Characteristics analysis of joint space inverse mass matrix for the optimal design of a 6- DOF parallel manipulator," *Mechanism and Machine Theory*, vol. 45, pp. 722-739, 2010.
- [3] Y. Su, B. Duan, and C. Zheng, "Genetic design of kinematically optimal fine tuning Stewart platform for large spherical radio telescope," *Mechatronics*, vol. 11, pp. 821-835, 2001.
- [4] Hao, F., and J-P. Merlet. "Multi-criteria optimal design of parallel manipulators based on interval analysis." *Mechanism and machine theory* 40.2 (2005): 157-171.
- [5] Taherifar, A., et al. "Kinematic Optimization of Stewart Platform for Simulators Using Genetic Algorithm."
- [6] Z. Gao, D. Zhang, Y. Ge, Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches, *Robot. Comput.-Integr. Manuf.* 26 (2010) 180–189.
- [7] Neto, Clodoaldo Schutel Furtado, and Clodoaldo Schutel. "Orientation workspace optimization for a 6-RUS parallel robot." (2014).
- [8] Mirshekari, Erfan, Afshin Ghanbarzadeh, and Kourosh Heidari Shirazia. "Structure Comparison and Optimal Design of 6-RUS Parallel Manipulator Based on Kinematic and Dynamic Performances." *Latin American Journal of Solids and Structures* 13.13 (2016): 2414-2438.
- [9] J. Denavit, R.S. Hartenberg, A kinematic notation for lower pair mechanism based on matrices, *J. Appl. Mech.* 22 (1955) 215–221.
- [10] W. Khalil, J. Kleinfinger, A new geometric notation for open and closed-loop robots, in: *Proceedings of the IEEE conference on robotics automation*: 1174–1180, 1986.
- [11] Merlet, Jean-Pierre. *Parallel robots*. Vol. 128. Springer Science & Business Media, 2006.
- [12] Bonev, Ilian Alexandrov. *Geometric analysis of parallel mechanisms*. Canada: Université Laval, 2002.
- [13] P. Paul, *Robot manipulators: mathematics, programming and control*, MIT Press, Cambridge, 1981.
- [14] Taghirad, Hamid D. *Parallel robots: mechanics and control*. CRC press, 2013.
- [15] Angeles, Jorge. *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Vol. 124. Springer Science & Business Media, 2013.
- [16] Lara-Molina, F. A., J. M. Rosario, and Didier Dumur. "Multi-objective optimization of stewart-gough manipulator using global indices." *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*. IEEE, 2011.
- [17] Babu, Sabbavarapu Ramana, et al. "Design for optimal performance of 3-RPS parallel manipulator using evolutionary algorithms." *Transactions of the Canadian Society for Mechanical Engineering* 37.2 (2013): 135.
- [18] J. H. Holland, *Adaptation in Natural and Artificial Systems*. The MIT press Cambridge, MA, USA, 1992.
- [19] J. S. Arora. *Introduction to Optimum Design*. Elsevier Academic Press, 2 edition, 2004.
- [20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948.
- [21] Q. Xu and Y. Li, Error analysis and optimal design of a class of translational parallel kinematic machine using particle swarm optimization, *J. Robotica* 27, 67–78 (2009).
- [22] D. ping Tian, "A Review of Convergence Analysis of Particle Swarm Optimization," *International Journal of Grid and Distributed Computing*, vol. 6, no. 6, pp. 117–128, 2013.
- [23] Gosselin, C. M., Wang, J., (2000). Static balancing of spatial six degree of freedom parallel mechanisms with revolute actuators. *Journal of Robotic Systems* 17(3): 159-170.

Authors' Profiles



Alaa Aldeen JOUMAH is student in the Master of Robotics and Control, specialty: Robotics at the Higher Institute for Applied Sciences and Technology (HIAST), Damascus, Syria. He received his B.S. degree in Mechatronics at HIAST in 2013.



Chadi ALBITAR received his B.S. degree in Mechatronics from the Higher Institute for Applied Sciences and Technology (HIAST), Damascus, Syria in 2000, the M.S degree in Photonics, Image and Cybernetics, specialty: Robotics and Vision from the National Higher School of Physics of Strasbourg (ENSPS), France in 2004, and the Ph.D. degree in Vision and Robotics from the University of Strasbourg, France in 2009. He is currently Head of Automation Lab and an associate Professor in the department of Mechanical and Electronic Systems at HIAST.

How to cite this paper: Alaa Aldeen Joumah, Chadi Albitar, "Design Optimization of 6-RUS Parallel Manipulator Using Hybrid Algorithm", *International Journal of Information Technology and Computer Science (IJITCS)*, Vol.10, No.2, pp.83-95, 2018. DOI: 10.5815/ijitcs.2018.02.08